



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

**PROGRAMOVÉ VYBAVENÍ PRO ZPRACOVÁNÍ
VÍCEROZMĚRNÝCH DAT ZE SENZORŮ PLYNŮ**

SOFTWARE FOR THE PROCESSING OF MULTIDIMENSIONAL DATA FROM GAS SENSORS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Denys Almazov

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Adámek, Ph.D.

BRNO 2019

Bakalářská práce

bakalářský studijní obor **Mikroelektronika a technologie**

Ústav mikroelektroniky

Student: Denys Almazov

ID: 186487

Ročník: 3

Akademický rok: 2018/19

NÁZEV TÉMATU:

Programové vybavení pro zpracování vícerozměrných dat ze senzorů plynů

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku zpracování a shlukování vícerozměrných dat. Dále nastudujte programování v jazyce Python a Wiring. Stručně se seznámte s vývojovou platformou Arduino a připojením plynových chemorezistivních senzorů přes tuto platformu.

Vyberte optimální metodu pro zpracování a shlukování vícerozměrných dat z plynových chemorezistivních senzorů a navrhnete základní kroky pro jejich použití v prototypu elektronického nosu. Realizujte tyto kroky a otestujte aplikaci na vybraných vzorcích vůní. Zhodnoťte Váš návrh a jeho možnosti.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 4.2.2019

Termín odevzdání: 30.5.2019

Vedoucí práce: Ing. Martin Adámek, Ph.D.

Konzultant:

doc. Ing. Jiří Háze, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato bakalářská práce se věnuje realizaci a testování programového vybavení pro zpracování vícerozměrných dat ze senzorů plynů. Práce seznamuje čtenáře s některými metodami zpracování vícerozměrných dat. Součástí práce je popis a použití metody PCA, popis zařízení určeného k detekci zápachů nebo chutí – elektronického nosu, realizace a testování programu pro předzpracování a zpracování dat. Program pro předzpracování dat je schopen zobrazovat data ze zařízení na grafu v reálném čase a uložit je do souboru. Program pro zpracování dat umožňuje nahrát data ze souboru, snížit jejich rozměrnost a zobrazit výsledky do 2D grafu. Tento program byl otestován na jednoduchém elektronickém nosu, který obsahuje tři senzory.

Klíčová slova

Elektronický nos, senzory plynů, PCA algoritmus, Wiring, Python.

Abstract

This work deals with realization and testing software for processing multidimensional data from a gas sensor. This work introduces readers to some multidimensional data processing method. The part of work is a description and usage PCA method, description of the smell or taste detection device – electronic nose, realisation and testing of the program for preprocessing and processing data. Preprocessing data program can visualise data from device on the graf in real time and save it to the file. Data processing program can read data from file, decrease their dimensionality and visualize results in 2D graf. This program was tested on a simple electronic nose that contains three sensors.

Keywords

Electronic nose, gas sensors, PCA algorithm, Wiring, Python.

Bibliografická citace:

ALMAZOV, Denys. Programové vybavení pro zpracování vícerozměrných dat ze senzorů plynů. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/119429>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce Martin Adámek.

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Martinu Adámkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování bakalářské práce.

Obsah

1.	Úvod	10
2.	Nejčastěji využívané metody pro zpracování dat ze senzorů plynů	11
3.	Redukce dimenzí pomocí PCA	13
3.1	Úvod do PCA analýzy	13
3.2	Směr s maximálním rozptylem	14
3.3	Klíčové kroky PCA	15
4.	Elektronický nos	18
5.	Použité počítačové jazyky a vybavení	20
5.1	Jazyk Wiring	20
5.2	Jazyk Python	21
5.3	Mikrokontrolér ATmega 2560	22
5.4	Jednotka předběžného zpracování	22
6.	Návrh a realizace programového vybavení pro analýzu PCA	24
6.1	Základní požadavky na využití zařízení	24
6.2	Návrh blokového schématu pro zpracování dat z elektronického nosu	24
6.3	Programování mikrokontroléru	25
6.4	Realizace programu pro předzpracování dat pomocí Pythonu.	26
6.4.1	PySerial	26
6.4.2	Uživatelské rozhraní	26
6.4.3	Zobrazování grafu v reálném čase pomocí matplotlib	28
6.4.4	Výsledný blokový diagram programu pro předzpracování dat.	29
6.5	Realizace a testování programu pro zpracování dat pomocí PCA	31
6.5.1	Knihovny Pandas, Scikit-learn, matplotlib jako základ programu ...	31
6.6	Testování programu zpracování dat na reálných datech	34
6.7	Výhody a nevýhody použitého řešení	35
7.	Závěr	36

Seznam symbolů a zkratk

Zkratky:

PCA	...	<i>Principal component analysis</i> – analýza hlavních komponent
LDA	...	<i>Latent Dirichlet allocation</i> – latentní dirichletova alokace
CA	...	<i>Correspondence analysis</i> – korespondenní analýza
HCA	...	<i>Hierarchical clustering</i> – shluková analýza
PLSR	...	<i>Partial least squares regression</i> – částečná regrese nejmenších čtverců
ANN	...	<i>Artificial neural networks</i> – umělá neuronová síť
DFA	...	<i>Discriminant function analysis</i> – diskriminační analýza
MLP	...	<i>multilayer perceptron</i> – vícevrstvý perceptron
NN	...	<i>Neural networks</i> – neuronová síť
MSI	...	<i>Multispectral imaging</i> – multispektrální snímkování
GA	...	<i>Genetic algorithm</i> – genetický algoritmus
k-NN	...	<i>k -nearest neighbor</i> – algoritmus k-nejbližších sousedů
SVM	...	<i>Support vector machines</i> – metoda podpůrných vektorů

Symboly:

V	...	rozptyl
μ_d	...	střední hodnota
σ_d	...	směrodatná odchylka
\mathbf{B}	...	matice, která obsahuje vlastní vektory
\mathbf{b}	...	vlastní vektor kovarianční matice
S	...	kovarianční matice
λ	...	Lagrangeův multiplikátor
\mathfrak{L}	...	Lagrangian

Seznam obrázků

Obrázek 2.1: Výsledky PCA pro vzorky mletého hovězího masa, které byly na základě údajů MSI pomíchané s vepřovým masem, Ropodi et al. [10].....	12
Obrázek 3.1: Datová sada se souřadnicemi x_1 a x_2 , M. P. Deisenroth et al. [19].....	13
Obrázek 3.2: Komprimovaná datová množina, kde je pouze souřadnice x_1 , M. P. Deisenroth et al. [19]	13
Obrázek 3.3: PCA hledá projekci dat na podprostor, ve kterém je rozptyl největší, M. P. Deisenroth et. al. [19]	14
Obrázek 3.4: Kroky PCA, M. P. Deisenroth et. al. [19].....	17
Obrázek 4.1: Základní blokové schéma elektronického nosu	19
Obrázek 5.1: Arduino IDE.....	21
Obrázek 5.2: Pohled na základní desku Arduino MEGA 2560	23
Obrázek 6.1: Systémový blokový diagram.....	24
Obrázek 6.2: Blokový diagram pro zpracování dat	25
Obrázek 6.3: Dialogové okno pro zvolení sériového portu.....	27
Obrázek 6.4: Dialogové okno pro zvolení počtů senzorů	27
Obrázek 6.5: Hierarchie objektů matplotlib	28
Obrázek 6.6: Zjednodušený blokový diagram programu pro předzpracování dat.....	30
Obrázek 6.7: Testování funkčnosti programů pro předzpracování dat	31
Obrázek 6.8: Výsledný bodový graf po zpracování dat	35

Seznam tabulek

Tabulka 2.1: Potraviny zkoumané elektronickým nosem a pro ně využívané metody zpracování dat	12
Tabulka 4.1: Použité plynové senzory v práci Estakhroveh et al. (2018). Zpracováno podle Estakhroyeh et al. [23]	19
Tabulka 6.1: Statistické funkce knihovny Pandas.....	31
Tabulka 6.2: Prvních pět dat ze senzorů před standardizací (vlevo) a po standardizaci (vpravo)	32
Tabulka 6.3: Příklad dat po převedení do hlavních komponent	33
Tabulka 6.4: Seznam použitých vzorků	34

1. ÚVOD

Čich je unikátní smysl. Bez něj si lze sotva představit komfortní život na Zemi. Díky čichu je člověk schopen chránit se před konzumací zkažených potravin a před některými jedovatými plyny, jako jsou například H_2S a NO_2 . Bohužel kvůli fyziologickým faktorům lidského organismu je schopnost identifikovat vůně a pachy u člověka velmi omezená. Průměrný člověk má kolem 5 milionů čichových buněk, zatímco pes má 220 milionů čichových receptorů [1]. Pouze tato skutečnost je již dostačující pro představu o tom, jak je lidský čich relativně špatně vyvinutý. Lidstvo bylo sice schopno dosáhnout významného úspěchu ve výcviku psů, avšak komunikace s nimi vytváří omezení ve využívání jejich nejsilnější schopností. Obdobně je to i ve výcviku jiných živočišných druhů (prase divoké, včela domácí).

Elektronická identifikace opravdu velkého množství vůní by mohla ve velké míře zlepšit kvalitu a délku života člověka. Bylo by možné dříve objevit škodliviny v atmosféře, diagnostikovat poruchy různých zařízení (např. ve vytápění, spalování atd.), provádět prvotní sken různých nemocí. Takové perspektivy inspirují mnoho vědců a inženýrů k vyvinutí zařízení pro identifikaci vůní a pachů – elektronického nosu. Elektronický nos je radikálně odlišný od analyzátorů plynů. Ty mohou rozlišit pouze určité látky, ale nosy jsou zařízení pro rozpoznání široké skupiny chemikálií. Mohou nejen rozlišit známé vůně, ale jsou schopny se učit rozpoznávat nové pomocí strojového učení.

Z důvodu schopnosti identifikovat velké množství různých vůní musí elektronický nos obsahovat různé typy senzorů. Každý senzor však zvyšuje rozměrnost dat. Kvůli tomu jsou vizualizace a zpracování dat ze senzorů obtížné, ale existují matematické metody, pomocí kterých se lze s tímto problémem vypořádat. Hlavním cílem této práce je najít a realizovat optimální algoritmus zpracování dat pro jednoduchý elektronický nos.

Teoretická část práce se zabývá používanými metodami zpracování dat v potravinářském průmyslu a dále pojednává o tom, jakým způsobem lze vícerozměrná data redukovat. V práci byly rovněž popsány principy fungování elektronického nosu a jeho konstrukce včetně senzorů, které se používají.

Praktická část obsahuje návrh blokového schématu pro zpracování dat z elektronického nosu a popis realizace programů pro předzpracování a zpracování dat. Program pro zpracování dat byl otestován s využitím elektronického nosu, který obsahoval tři senzory.

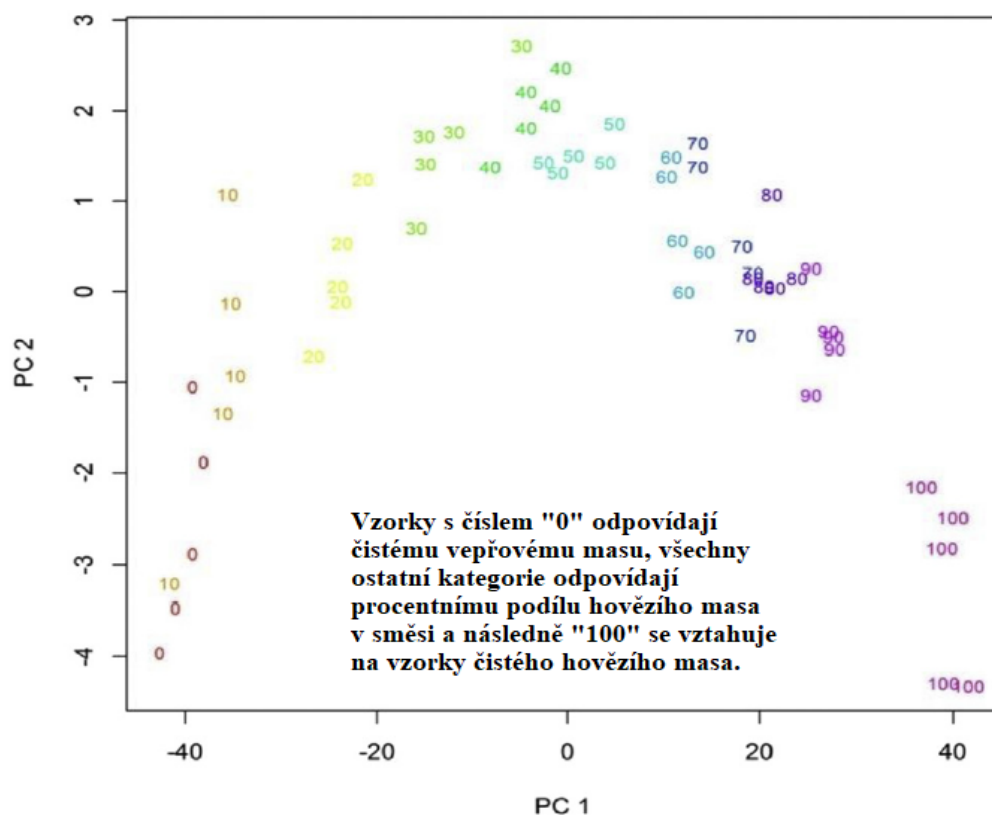
2. NEJČASTĚJI VYUŽÍVANÉ METODY PRO ZPRACOVÁNÍ DAT ZE SENZORŮ PLYNŮ

Gutierrez-Osuna [2] rozdělil proces zpracování a analýzy dat na následující bloky: předběžné zpracování signálu, snížení dimenze dat, klasifikace, regrese, shlukování a ověřování. Ve své práci ukázal, jak aplikační oblast a typy snímačů ovlivňují výběr optimálních metod zpracování dat. Předběžné zpracování signálů je zapotřebí k vylepšení a optimalizaci dat, která budou potřebná k další diskriminační analýze [3]. Pokud chceme snížit dimenzi dat, PCA (neboli analýza hlavních komponent) je neocenitelným nástrojem pro vizualizaci datových struktur, která je pro klasifikační problém s unimodální hustotou pravděpodobnosti LDA (neboli latentní Dirichletovou alokací) "zlatým standardem". T. Dymerski et al. [4] ve své práci pro klasifikaci různých druhů medu použili tři typy analýz: PCA, LDA a CA (neboli shluková analýza). Pomocí LDA pak identifikovali všech pět druhů medu, pomocí PCA a CA pouze tři. Pro shlukování dat lze použít HCA (neboli hierarchickou shlukovou analýzu), jako to udělali S. Kiani et al. [3] a J. M. Kalaw et al. [5], aby ověřili výsledky PCA. Výsledek HCA je zobrazen jako dendrogram, který ukazuje všechny kroky hierarchického postupu. Jinou často používanou metodou shlukování je metoda shlukování „k-means“, kterou úspěšně použili ve své práci M. F. Rutolo et al. [6].

HCA a PCA se používají v potravinářských aplikacích nejčastěji [7], [8]. V případě HCA je nejprve vypočítána vzdálenost (euklidovská nebo jiná) mezi objekty (vzorky), která je dále zpracována pomocí aglomeračního algoritmu vzdálenosti za účelem vytvoření dendrogramu. Ten kombinuje vzorky podle jejich vzájemných podobností [9]. Naproti tomu PCA je široce používaná metoda, která umožňuje výrazné snížení dimenze a současně volí nekorelované proměnné (nazývané počítací s hlavními součástmi), které vysvětlují většinu odchylek v datové sadě. Hlavní výhodou této metody je, že může překládat multivariační datovou množinu do datové sady několika hlavních součástí a proto pomáhá vizualizovat vzorky. Například hodnoty jednotlivých vzorků se v PCA vykreslují na základě jejich podobností nebo odlišností obvykle pomocí 2D nebo 3D projekce těchto vzorků. Díky této schopnosti je PCA obvykle první analýzou provedenou při prezentaci s tímto typem dat. Dále může být použita pro detekci odlehlých hodnot a pro zvýraznění důležitých vlastností, které mají být využity pro další analýzu. Tento proces je známý jako variabilní výběr [7].

Na obrázku 2.1 je PCA aplikována na data MSI (Multispectral images) ze vzorků mletého hovězího masa, které bylo smíseno s vepřovým masem [10]. Vzorky označené číslem "0" odpovídají čistému vepřovému masu, všechny ostatní kategorie odpovídají procentuální hmotnosti hovězího masa v směsi a následně "100" se vztahuje na vzorky čistého hovězího masa. Z výsledků je zřejmé, že PCA dokáže vizualizovat vícerozměrné pole (v tomto případě s celkem 36 proměnnými) v 2D grafu. To poukazuje na dostatečnou

míru diskriminace mezi vzorky s různým zastoupením hovězího a vepřového masa a tím i na možnost odhalování falšování těchto směsí



Obrázek 2.1: Výsledky PCA pro vzorky mletého hovězího masa, které byly na základě údajů MSI pomíchané s vepřovým masem, Ropodi et al. [10]

Elektronický nos je zařízení určené k detekci zápachů nebo chutí, proto je často využívaným přístrojem v potravinářství. Tabulka 2.1 obsahuje seznam potravin analyzovaných pomocí elektronického nosu a pro ně odpovídajících metod zpracování dat.

Tabulka 2.1: Potravin y zkoumané elektronickým nosem a pro ně využit é metody zpracování dat

Potravina	Metody zpracování dat	Odkazy
Med	PCA, LDA, CA	T. Dymerski et al. [4]
Jahodový džus	LDA, PLSR	Qiu et al. [11]
Mango	PCA, LDA	Zakaria et al. [12]
Sumec filet	PCA, ANN	Wilson et al. [13]
Hovězí filety	PCA-DFA	Papadopoulou et al. [14]
Stolní olivy	PCA, HCA, DFA, MLP-NN	Panagou et al. [15]
Rajčata	PCA	Concina et al. [16]
Jahoda	PCA, ANN	Pan et al. [17]
Broskev	PCA, LDA, PCR, PLSR	Zhang et al. [18]

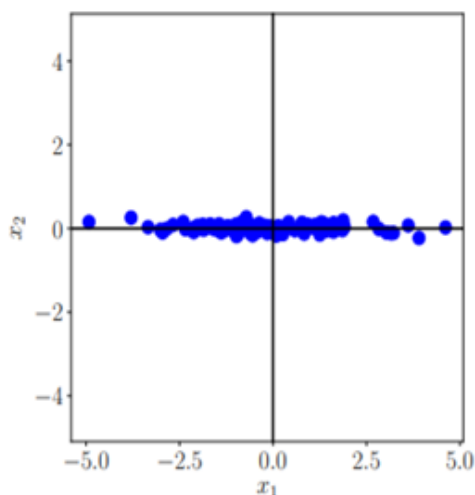
3. REDUKCE DIMENZÍ POMOCÍ PCA

3.1 Úvod do PCA analýzy

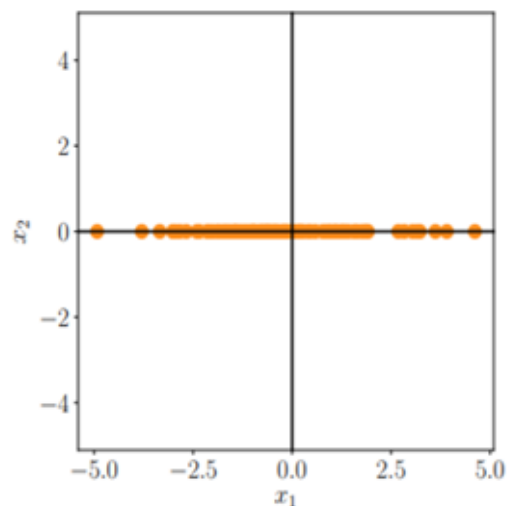
Při práci s vysoce dimenzionálními daty vznikají určité problémy: data je složité analyzovat, jejich interpretace je obtížná, vizualizace je prakticky nemožná a z praktického hlediska může být uchování vícerozměrných datových vektorů náročné a drahé. Vícerozměrná data jsou však často nadbytečná, tzn. mnoho rozměrů je redundantních a může být interpretováno kombinací ostatních dimenzí. Navíc rozměry ve vysoce dimenzionálních datech jsou často korelovány tak, že data mají vnitřní strukturu s nižším rozměrem. Proces snižování dimenze využívá tuto strukturu a korelaci a umožňuje pracovat s kompaktnější reprezentací dat, v ideálním případě bez ztráty informací. Z tohoto důvodu je možné uvažovat o snížení dimenze jako o kompresní technice, podobně jako je tomu u obrázku formátu .jpeg nebo hudebním souboru typu .mp3.

Při procesu snížení počtu rozměrů je obecně využíváno vlastností vysoce dimenzionálních dat, které často leží na nízko dimenzionálním podprostoru (např. obrázky).

Obrázek 3.1 uvádí názorný příklad ve dvou rozměrech. Variace dat ve směru osy x_2 je nízká, proto mohou být datové body vyobrazeny na ose x_1 s minimální ztrátou informací. Pro popis dat na obrázku 3.2 je nutná pouze souřadnice x_1 a data leží v jednorozměrném podprostoru R^2 .



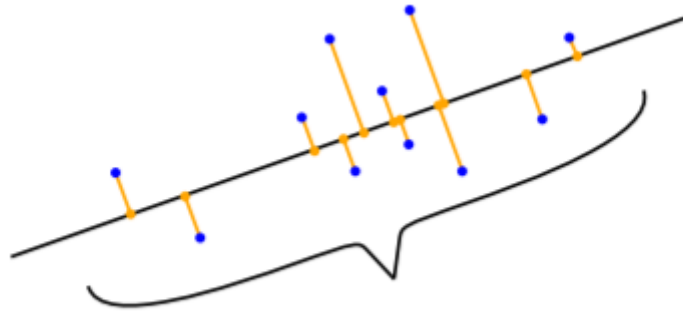
Obrázek 3.1: Datová sada se souřadnicemi x_1 a x_2 , M. P. Deisenroth et al. [19]



Obrázek 3.2: Komprimovaná datová množina, kde je pouze souřadnice x_1 , M. P. Deisenroth et al. [19]

V PCA je zájem o nalezení projekcí \tilde{x} datových bodů x_n , které jsou stejně podobné původním datovým bodům, ale které mají podstatně nižší rozměrnost. Obrázek 3.1 zobrazuje příklad snížení rozměrnosti. Na obrázku 3.2 je ignorována x_2 souřadnice, protože neobsahovala příliš mnoho informací, takže komprimovaná data jsou podobná původním datům na obrázku 3.1. Bylo by možné se rozhodnout ignorovat souřadnici x_1 , ale pak by byly komprimované údaje velmi odlišné od původních dat a mnoho informací v datech by bylo ztraceno.

Rozptyl je indikátorem šíření dat a může být v PCA použit jako algoritmus redukce dimenzionality, který maximalizuje rozptyl v nízkorozměrném zobrazení dat tak, aby bylo uchováno co nejvíce informací. PCA tak hledá projekci dat na podprostor, ve kterém je rozptyl největší [19]. Toto znázorňuje obrázek 3.3.



Obrázek 3.3: PCA hledá projekci dat na podprostor, ve kterém je rozptyl největší, M. P. Deisenroth et. al. [19]

3.2 Směr s maximálním rozptylem

Cílem je maximalizovat rozptyl první koordináty z_1 z $\mathbf{z} \in \mathbb{R}^M$

$$V_1 := \frac{1}{N} \sum_{n=1}^N z_{1n}^2 \quad (1)$$

z_{1n} je definována jako první souřadnice nízkodimenzionální reprezentace $\mathbf{z}_n \in \mathbb{R}^M$ z $\mathbf{x}_n \in \mathbb{R}^D$.

První komponenta ze \mathbf{z}_n je dána jako:

$$\mathbf{z}_{1n} = \mathbf{b}_1^\top \mathbf{x}_n, \quad (2)$$

Dosazením (2) do (1) vychází

$$V_1 = \frac{1}{N} \sum_{n=1}^N (\mathbf{b}_1^\top \mathbf{x}_n)^2 = \frac{1}{N} \sum_{n=1}^N \mathbf{b}_1^\top \mathbf{x}_n \mathbf{x}_n^\top \mathbf{b}_1 \quad (3a)$$

$$= \mathbf{b}_1^\top \left(\frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top \right) \mathbf{b}_1 = \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1, \quad (3b)$$

kde \mathbf{S} je kovarianční matice. Tuto matici lze formulovat jako

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top, \quad (4)$$

tedy skalární součin dvou vektorů, který je symetrický vzhledem k jejím argumentům (proto v rovnici 3a je $\mathbf{x}_n^\top \mathbf{b}_1 = \mathbf{b}_1^\top \mathbf{x}_n$). Libovolné zvýšení velikosti vektoru \mathbf{b}_1 zvyšuje V_1 , tj. vektor \mathbf{b}_1 , který má dvojnásobnou velikost, může mít za následek, že V_1 bude mít potenciálně čtyřikrát větší velikost. Proto jsou omezována všechna řešení $\|\mathbf{b}_1\|^2 = 1$. To vede k redukci problému optimalizace, ve kterém je hledán směr, podle kterého se data nejvíce liší.

S omezením prostoru pro řešení na jednotkové vektory je možné nalézt vektor \mathbf{b}_1 , který ukazuje směr maximálního rozptylu

$$\begin{aligned} \max \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1 \\ \|\mathbf{b}_1\|^2 = 1 \end{aligned} \quad (5)$$

Je získán Lagrangian

$$\mathfrak{L}(\mathbf{b}_1, \lambda) = \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1 + \lambda_1 (1 - \mathbf{b}_1^\top \mathbf{b}_1) \quad (6)$$

Parciální derivace \mathfrak{L} vůči \mathbf{b}_1 a λ_1 jsou

$$\frac{\partial \mathfrak{L}}{\partial \mathbf{b}_1} = 2\mathbf{b}_1^\top \mathbf{S} - 2\lambda_1 \mathbf{b}_1^\top, \quad \frac{\partial \mathfrak{L}}{\partial \lambda_1} = 1 - \mathbf{b}_1^\top \mathbf{b}_1, \quad (7)$$

Položením těchto derivací rovných 0 obdržíme vztah

$$\mathbf{S} \mathbf{b}_1 = \lambda_1 \mathbf{b}_1, \quad (8)$$

$$\mathbf{b}_1^\top \mathbf{b}_1 = 1 \quad (9)$$

Porovnáním (3b), (8) a (9) lze získat

$$\mathbf{V}_1 = \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1 = \lambda_1 \mathbf{b}_1^\top \mathbf{b}_1 = \lambda_1 \quad (10)$$

Vektor \mathbf{b}_1 je vlastní vektor kovarianční matice \mathbf{S} a Lagrangeův multiplikátor λ_1 hraje roli vlastního čísla. Rozptyl dat promítnutých na jednorozměrný podprostor se rovná vlastní hodnotě, která je spojena se základním vektorem \mathbf{b}_1 , který se rozprostírá v tomto podprostoru. Proto, aby byl maximalizován rozptyl, je volen základový vektor spojený s největší vlastní hodnotou datové kovarianční matice. Tento vlastní vektor se nazývá první hlavní složkou (principal component – dále PC).

3.3 Klíčové kroky PCA

M. P. Deisenroth et al. [19] ve své knize "Mathematics For Machine Learning" popsali klíčové kroky PCA na praktickém příkladu uvedeném na obrázku 3.4. Tyto klíčové kroky zahrnují:

1. **Střední odčítání.** Data jsou centrována výpočtem střední hodnoty μ datové sady, která je odečtena od každého jednotlivého datového bodu. To zajišťuje, že datová sada má střední hodnotu 0. Odečtení střední hodnoty není nutné, ale snižuje riziko číselných chyb.
2. **Standardizaci.** Účelem je rozdělit datové body na standardní odchylku σ datové sady pro každý rozměr $d = 1, \dots, D$. Údaje jsou nyní bez jednotek a mají rozptyl 1 na každé ose, jak je naznačeno dvěma standardizačními šipkami na obrázku 3.4(c).
3. **Správný rozklad kovarianční matice.** Je nutné vypočítat kovarianční matici, její vlastní hodnoty a odpovídající vlastní vektory. Vzhledem k tomu, že kovarianční matice je symetrická, vlastní vektory tvoří ortogonální základ. Na obrázku 3.4(d) jsou vlastní vektory zmenšeny podle velikosti příslušné vlastní hodnoty. Dlouhý vektor pokrývá hlavní podprostor. Údaje kovarianční matice jsou reprezentovány elipsou.
4. **Projekce.** Je možno projektovat jakýkoli datový bod $x_* \in \mathbb{R}^D$ do hlavního podprostoru. Pro správné provedení je nutné standardizovat x_* pomocí střední hodnoty μ_d a směrodatné odchylky σ_d dat v dimenzi d .

$$x_*^d \leftarrow \frac{x_*^d - \mu_d}{\sigma_d}, \quad d = 1, \dots, D, \quad (11)$$

kde x_*^d je d -tá komponenta x_* . Je získána projekce

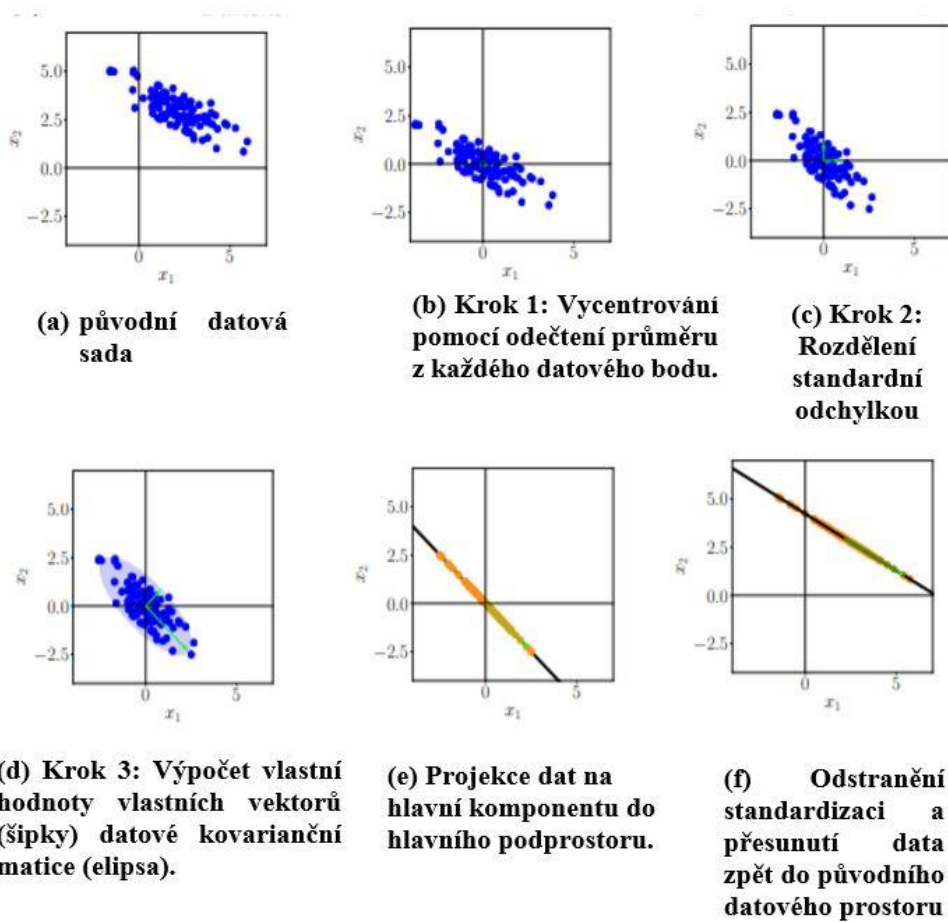
$$\widetilde{x}_* = BB^T x \quad (12)$$

se souřadnicemi

$$z_* = B^T x \quad (13)$$

Zde B je matice, která obsahuje vlastní vektory, které jsou spojeny s největšími vlastními hodnotami kovarianční matice dat ve formě sloupců. PCA vrací souřadnice (13) nikoliv projekce x_* . Pro získání projekce v původním datovém prostoru (tj. před standardizací) je nutné vrátit normalizaci (11) a násobit směrodatnou odchylkou před přidáním střední hodnoty.

$$\widetilde{x}_*^d \leftarrow \widetilde{x}_*^d \sigma_d + \mu_d, \quad d = 1, \dots, D. \quad (14)$$



Obrázek 3.4: Kroky PCA, M. P. Deisenroth et. al. [19]

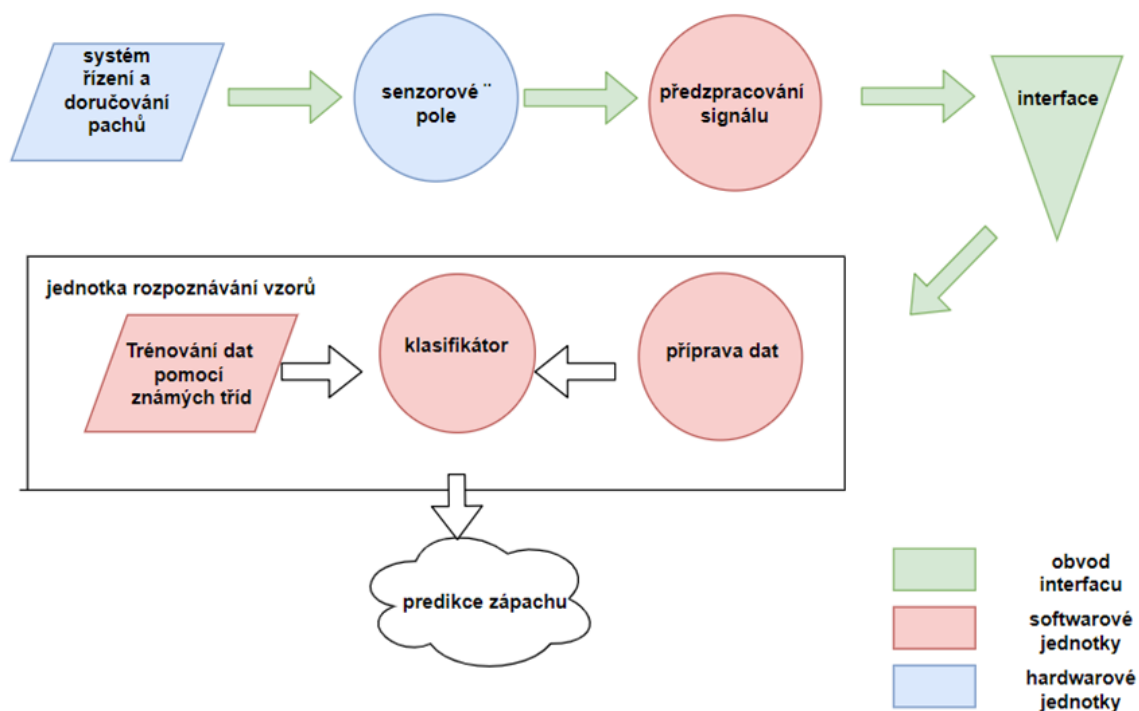
4. ELEKTRONICKÝ NOS

Systémy elektronického nosu (dále E-nos) se skládají z hardwarových a softwarových součástí. Hardwarové části E-nosu jsou soustavy plynových snímačů a elektromechanické systémy pro manipulaci a dodávku pachů. Dosud byly vyvinuty různé typy senzorů pro pole snímačů E-nosů. Oxid cínu je jedním z nejběžnějších oxidů kovů, který vykazuje vysokou citlivost vůči širokému spektru organických výparů [20]. Konvalina et al. [21] ve své práci používali plynové sensory vyrobené nanomateriály pro dechové testy a diagnostiky nemoci. V jiné studii Tisch et al. [22] představili spektrum chemických senzibilních kovových nanočástic (MCNPs), které jsou chemicky citlivé na diagnostiku onemocnění založené na dechovém testování.

V elektronických aplikacích E-nosů se používají různé matrice senzorů k vytvoření reaktoru, který reaguje odlišně na širokou škálu chemických plynů. Stejně jako obecný nos, E-nos musí mít manipulační systém, který slouží pro doručení cílového pachu do senzoru plynu. K přenosu molekuly zápachu do komory snímače E-nosu mohou být použity různé techniky odběru vzorků, jako je difúze plynu, různé předkoncentrátory a bublinkery.

Softwarové části E-nosu zahrnují předzpracování naměřených dat a algoritmy pro rozpoznávání vzoru (pattern recognition, dále PR). Klasifikátor rozlišuje třídy zápachu. Pro předzpracování se často používají metody jako GA, PCA, LDA. Pro další zpracování jsou použity další metody, jako jsou k-NN, SVM, ANN, HCA.

Jak je znázorněno na obrázku 4.1, hlavní softwarové / hardwarové součásti E-nosu jsou: *systém řízení a transport pachů*, *senzorové pole*, *jednotka předběžného zpracování signálu* a *interface*. Systém řízení a transport pachů přivádí pachy do pole senzorů. Výstupní signály snímačů jsou předběžně zpracovávány a poté dodávány do jednotky rozpoznávání vzorů pro detekci zápachu pomocí komunikačního rozhraní. Jednotka rozpoznávání vzorů používá údaje se známými třídami pro určení třídy vzorků neznámého zápachu.



Obrázek 4.1: Základní blokové schéma elektronického nosu

Senzorová buňka je jednou z nejdůležitějších a neúčinnějších částí E-nosu. Snímače jsou vybrány na základě jejich přesnosti a citlivosti na různé plyny. Zatím byly v této oblasti vyrobeny různé komerční senzory s citlivostí na vybrané plyny. Ve své práci Estakhroyeh et al. [23] vyvinuli E-nos, který lze použít ve víceúčelových aplikacích. Tabulka 4.1 uvádí seznam použitých senzorů.

Tabulka 4.1: Použité plynové senzory v práci Estakhroyeh et al. (2018). Zpracováno podle Estakhroyeh et al. [23]

Název	Cílový plyn	Rozsah detekce
MQ 135	Amoniak, sulfid, toluen, vodík, kouř, benzen	10–300 ppm NH ₃ , 10–300 ppm alkohol, 10–1000 ppm benzen
MQ 2	Kouř / Hořlavý plyn	300~10000 ppm (hořlavý plyn)
MQ 3	Alkohol	10–1000 ppm
MQ 4	Methan / CH ₄ / LNG	300~10000 ppm
MQ 5	Propan	300~10000 ppm
MQ 6	LPG, propan, hořlavý plyn	Propan 100-10 000 ppm
MQ 7	Oxid uhelnatý (CO)	10~500 ppm
MQ 9	Oxid uhelnatý a methan	10~500 ppm CO, 300~10000 ppm CH ₄

5. POUŽITÉ POČÍTAČOVÉ JAZYKY A VYBAVENÍ

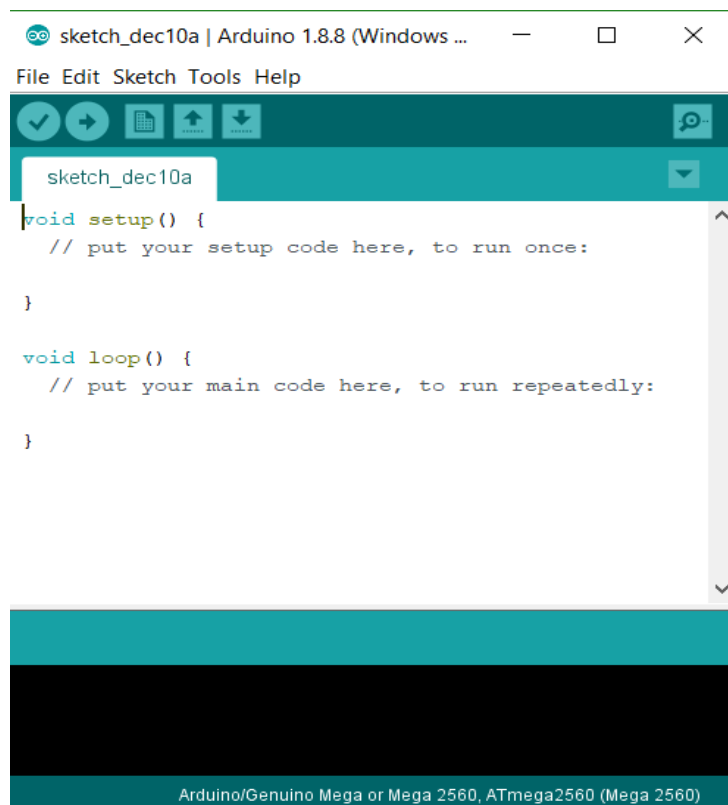
5.1 Jazyk Wiring

Wiring je open source jazyk/vývojový nástroj pro mikrokontroléry. Současné jádro AVR8 podporuje hardware Wiring a veškerý hardware založený na procesorech AVR atmega. Brzy budou k dispozici AVR Xmega, AVR Tiny, TI MSP430, Microchip PIC24 / 32 Series a STM M3 ARM Cores [24]. Pro programování v jazyce Wiring se používají Arduino IDE (obrázek 5.1) a Wiring IDE. Programy Wiring jsou napsány v C/C++. Pro spuštění programu je nutné definovat pouze dvě funkce [25]:

- `setup()` – funkce, která se spustí jednou na začátku programu a zpravidla obsahuje počáteční nastavení.
- `loop()` – automaticky opakovaně volaná funkce, když je deska připojena k napájení.

Některé další funkce Wiring [24]:

- `analogRead(pin)` – slouží ke čtení analogové hodnoty na vstupních pinech. Rozsah možných hodnot je 0-1023, kde 0 je 0 a 1023 je 5 voltů.
- `analogWrite(outpin, value)` – slouží k nastavení analogové hodnoty na pinu.
- `digitalRead(pin)` – čte stav digitálního pinu definovaného jako vstupní. Výsledek je HIGH / LOW.
- `digitalWrite(outpin, OUTPUT)` – zapíše na vybraný pin definovaný jako výstupní hodnotu HIGH, nebo LOW.



Obrázek 5.1: Arduino IDE

5.2 Jazyk Python

Pro vědce je důležitá dostupnost nástrojů pro efektivní numerickou a statistickou analýzu a vizualizaci objemných dat s využitím co nejjednodušších programovacích syntaxí. To je důvod, proč spousta z nich nepoužívá jazyky, jako jsou C, C++ a Java. Vzhledem k tomu, že tyto jazyky jsou funkčně bohaté, v některých případech nejsou schopny uspokojit potřeby rychlého učení, snadnosti používání a interaktivity s koncovým uživatelem díky meta-syntaxi a složitosti kódu [26].

Python je bezplatný, výkonný, univerzální a objektově orientovaný modulární programovací jazyk, který napsal Guido van Rossum. Byl široce akceptován akademickou a vědeckou komunitou díky jeho čistému a čitelnému kódovacímu stylu. Kód musí být odsazen pro definování funkcí, smyček a řídicích struktur [27]. To umožňuje jeho snadné čtení.

Shukla et al. [28] popsali ve své práci často využívané nástroje pro numerické zpracování:

- NumPy - Numerical Python je základní balíček pro numerické výpočty v Pythonu. Doplnuje vlastní typy dat Pythonu a umožňuje zpracovávat velké objemy dat rychle a efektivně. Má bohatou sadu funkcí pro práci s poli.
- SciPy - Scientific Python využívá N-rozměrné pole NumPy a nabízí řešení standardních problémů s výpočetními požadavky vědecké komunity. Kromě statistik se SciPy zabývá různými problémy, jako je numerická integrace, diferenciální rovnice, lineární algebra, tvarování a optimalizace křivek, digitální zpracování signálu, sparse matice atd.
- matplotlib - je balíček pro vykreslování 2D grafů. Umožňuje uživateli interaktivní vizualizaci dat.

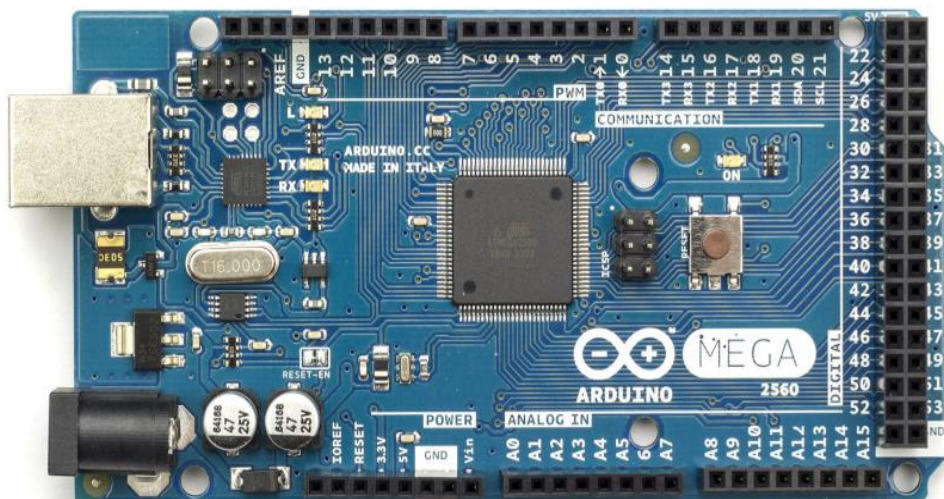
5.3 Mikrokontrolér ATmega 2560

ATmega2560 je nízkopříkonový CMOS 8 bit mikrokontrolér založený na vylepšené RISC architektuře. Jeho rychlost se přibližuje až 1 MIPS na MHz. Charakteristiky ATmega 2560:

- Rychlost CPU: 16 MHz
- 256 KBytů FLASH paměti programovatelné v systému
- 4 Kbyťů EEPROM paměti
- 8 Kbyťů Interní SRAM paměti
- 135 Výkonných instrukcí
- 32×8 pracovních registrů pro všeobecné použití
- Počet vstupů/výstupů: 86
- Napájecí Napětí Max: 5.5 V

5.4 Jednotka předběžného zpracování

Arduino MEGA 2560 (viz obrázek 5.2) může být použito v jakosti jednotky předběžného zpracování (Pre-processing Unit). Arduino MEGA 2560 bylo úspěšně použito v E-nosu od Estakhroveh et al. [23], který lze použít ve víceúčelových aplikacích. Arduino MEGA 2560 je open source a snadno se používá. Používá mikrokontrolér AT mega 2560 a má 54 digitálních vstupních a výstupních pinů, 4 UART, 16 analogových vstupů, konektor USB, konektor napájení a krystalový oscilátor 16 MHz. Tato deska je předprogramovaná se zavaděčem a kódy mohou být aktualizovány bez použití externího hardwarového programátoru.



Obrázek 5.2: Pohled na základní desku Arduino MEGA 2560

Pro implementaci funkce předběžného zpracování na vývojové desce Arduino se používá se jazyk C a IDE Arduino (viz obrázek 5.1). Tato deska obsahuje několik zařízení pro komunikaci s dalšími deskami, počítači nebo jinými mikrokontroléry a poskytuje čtyři hardwarové jednotky (UARTs) pro sériovou komunikaci TTL. Pro spuštění hardwarových jednotek je použita knihovna v C, která je otevřená a volně dostupná. Arduino Mega 2560 má 16 pinů pro analogové vstupy a může podporovat 16 plynových senzorů. Mega 2560 má 10bitový AD převodník. To znamená, že může mapovat vstupní napětí v rozsahu od 0 do 5 voltů do celočíselných hodnot od 0 do 1023. Čtení analogového vstupu trvá max. 260 μ s.

Obvykle se plynové snímače používají při vysokých teplotách, což vede k vysoké spotřebě energie pro senzorové pole. Externí zdroj je použitý pro nastavení pracovního bodu plynových senzorů. Deska může pracovat s externím napájením od 6 do 20 voltů [23].

6. NÁVRH A REALIZACE PROGRAMOVÉHO VYBAVENÍ PRO ANALÝZU PCA

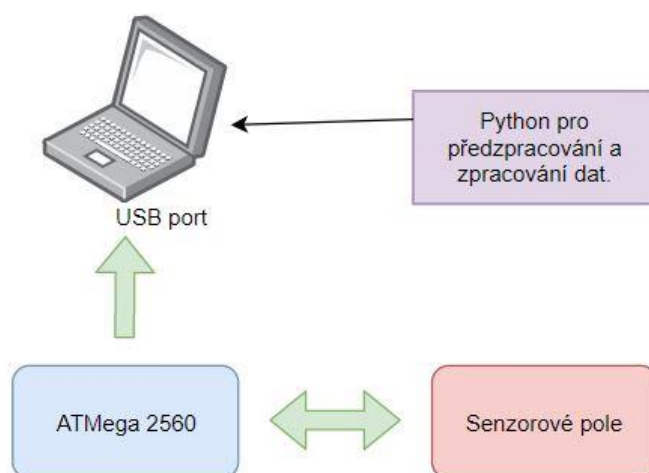
6.1 Základní požadavky na využití zařízení

Základním požadavkem je sběr dat ze sensorového pole. Musí být přítomna možnost jednoduchého připojení nového senzoru za účelem lepší práce algoritmu pro zpracování dat. Důležitá je i možnost rychlé a jednoduché implementace jiných algoritmů s odlišnou metodou zpracování dat v případě, že analýza PCA nebude schopna klasifikovat data. Proto se za základní požadavky mohou považovat:

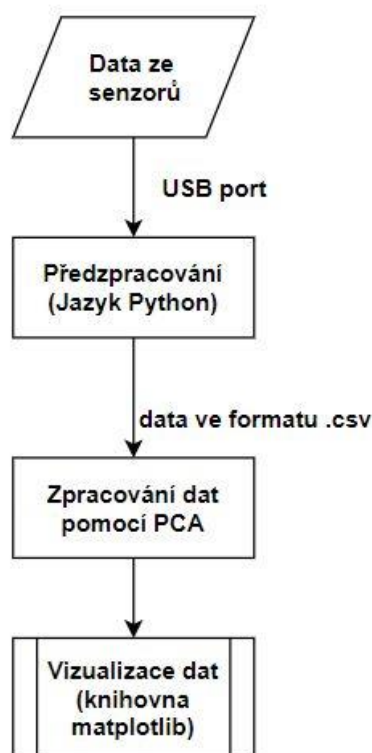
- jednoduchost zapojení,
- jednoduchost ovládání a programování,
- univerzálnost.

6.2 Návrh blokového schématu pro zpracování dat z elektronického nosu

Na obrázku 6.1 je znázorněn blokový diagram pro elektronický nos. Mikrokontrolér ATmega 2560 bude přijímat data z plynových senzorů a přes USB port posílat data do počítače. Tato komunikace bude nalaďena pomocí Wringu. Dále budou data předzpracována pomocí Pythonu a zapsána do souboru formátu .csv. Dalším krokem je spuštění Python skriptu, pomocí kterého budou data ze souboru zpracovávána a vizualizována (viz schéma na obrázku 6.2).



Obrázek 6.1: Systémový blokový diagram



Obrázek 6.2: Blokový diagram pro zpracování dat

6.3 Programování mikrokontroléru

Pro sbírání dat ze senzorů plynů je nutné nastavit piny na vstup. To bylo provedeno ve funkci `setup()`, která se spustí pouze jednou po připojení napětí. Nastavení pinů na vstup bylo provedeno pomocí funkce `pinMode()`. Příklad nastavení portu na vstup a čtení hodnoty:

```

void setup() {
    Serial.begin(9600);
    pinMode(A1, INPUT);
}

void loop() {
    double val1 = analogRead(A1);
    delay(100);
    sendToPC(&val1);
}
  
```

Ve funkci `loop()` se do proměnné `val1` zapisují data z analogových vstupů. Dále data posíláme do počítače pomocí funkce `sendToPC()`, ve které se data zapisují do bytového bufferu a poté je pomocí funkce `Serial.write()` celý buffer zapsán na sériový port. První

parametr funkce `write()` je buffer, druhý je jeho rozměr. Celý kód, který byl použit pro testování programu pro předzpracování, je obsažen v příloze 1.

```
void sendToPC(double* data1) {  
    byte* byteData1 = (byte*) (data1);  
    byte  buf[4]     =  {byteData1[0],  byteData1[1],  byteData1[2],  
byteData1[3]}  
    Serial.write(buf, 4);  
}
```

6.4 Realizace programu pro předzpracování dat pomocí Pythonu.

6.4.1 PySerial

PySerial umožňuje přístup k sériovému portu. Všechny soubory v knihovně jsou napsány v jazyce Python. PySerial podporuje Windows, OSX, Linux [29]. Příklad použití:

```
with serial.Serial('COM3', 9600, timeout=1) as ser:  
    x = ser.read()           # čtení jednoho bajtu  
    s = ser.read(10)         # čtení do deseti bajtu (timeout)  
    line = ser.readline()    # číst řádek
```

V programu se do sériového portu posílá bajtový buffer. Pro správné rozbalení dat byla použita funkce knihovny "struct" `struct.unpack(format, buffer)`. Pro parametr „format“ existují zkratky jako „f“ pro datový typ float, „d“ pro double, „i“ pro integer atd. [30].

6.4.2 Uživatelské rozhraní

Pro programování komponent uživatelského rozhraní byla zvolena knihovna pythonu EasyGUI. EasyGUI poskytuje snadno použitelné rozhraní pro jednoduchou interakci s GUI uživatelem. To nevyžaduje od programátora znalosti o tkinter, frames, widgets, callbacks nebo lambda. Všechny interakce grafického uživatelského rozhraní jsou vyvolány jednoduchými voláními funkcí, které vracejí výsledky [31]. V programu komponenty EasyGUI byly použity pro volání dialogových oken, kde si uživatel může vybrat sériový port, na kterém je připojeno zařízení, název grafiky a počet senzorů.

```
msg = "Select a port"  
title = "Choosing a serial port"  
port_name = easygui.choicebox(msg, title, ports)
```

```
graph_name = easygui.enterbox('Enter a name for the graph', '')
num_plots = easygui.integerbox('Enter the number of plots')
```

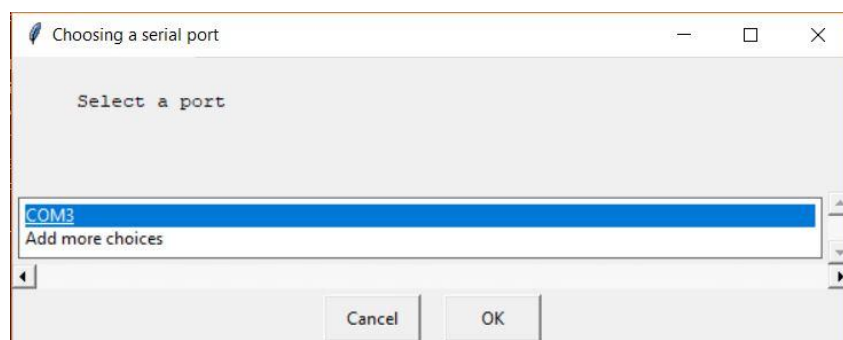
Pomocí `matplotlib.widget` bylo přidáno tlačítko "Start recording", které umožní uživatelům zvolit si, kdy začít zápis dat do souboru. Po zavření programu dojde k vyvolání dvou dalších dialogových oken, kde si uživatel může zvolit název souboru a cestu pro jeho uložení. Jeden soubor obsahuje všechna data ze senzorů, druhý obsahuje střední hodnoty pro každý senzor. Toto provedení bylo zvoleno proto, aby uživatel mohl kontrolovat kvalitu změřených hodnot a nemusel provádět další výpočty, pokud to není potřeba. Kód pro přidání tlačítka:

```
axes_button_add = pylab.axes([0.1, 0.9, 0.25, 0.075])
button_add = Button(axes_button_add, 'Start recording')
```

Parametry `[0.1, 0.9, 0.25, 0.075]` popořadě odpovídají za koordináty, šířku a výšku tlačítka. Dialogová okna pro zvolení cesty souboru se vyvolají takto:

```
path = easygui.filesavebox()
```

Příklady dialogových oken pro výběr portu a zvolení počtu senzorů jsou na obrázcích 6.3 a 6.4.



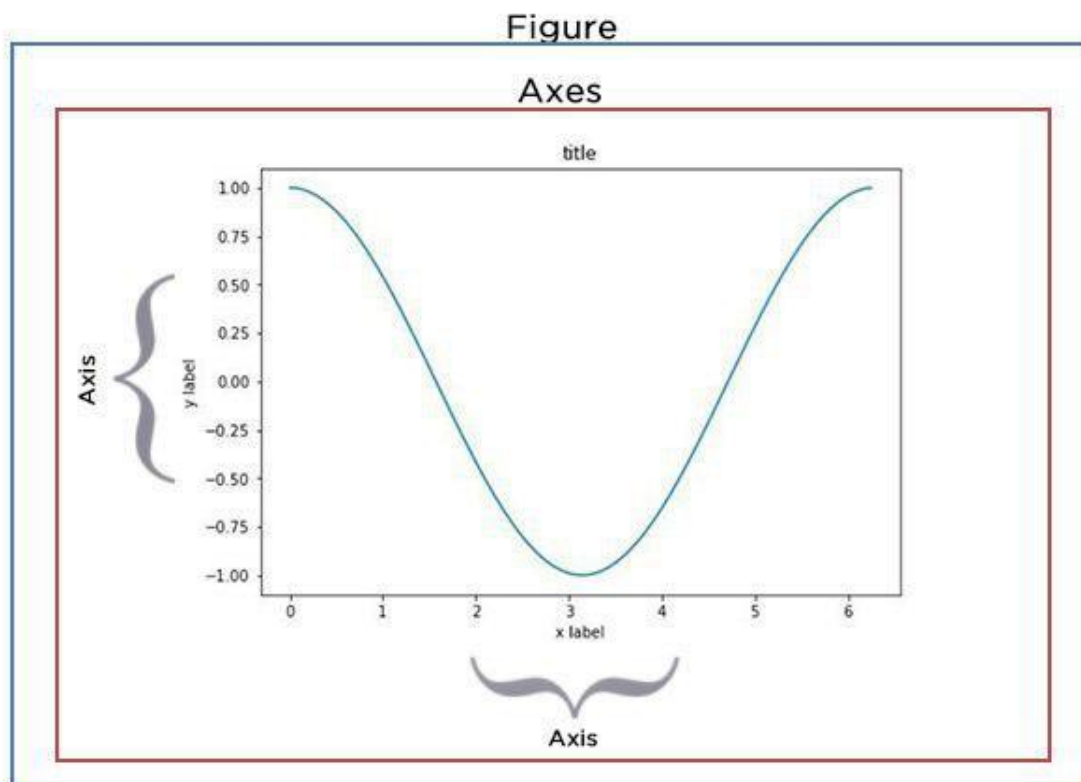
Obrázek 6.3: Dialogové okno pro zvolení sériového portu



Obrázek 6.4: Dialogové okno pro zvolení počtů senzorů

6.4.3 Zobrazování grafu v reálném čase pomocí matplotlib

Každý graf, který je vytvořen pomocí “matplotlib”, je založen na stromové struktuře objektů matplotlib. Objekt “Figure” je nejdůležitějším externím kontejnerem pro grafiku matplotlib, která může obsahovat více objektů “Axes”. Objekty “Axes” jsou transformovány na individuální grafy, nebo diagramy. Pod objekty Axes jsou v hierarchickém pořadí umístěny menší objekty, jako jsou jednotlivé čáry, značky, legendy a textové boxy. Prakticky každý “prvek” grafu je vlastní manipulovaný objekt Pythonu až po zkratky a značky [32]:



Obrázek 6.5: Hierarchie objektů matplotlib

Na začátku je nutné importovat balíčky:

```
import matplotlib.pyplot as plt
import matplotlib.animation as animation
```

Dalším krokem je vytvoření objektů “Figure”, “Axes” a “Axis”:

```
fig = plt.figure(figsize=(10, 8))
ax = plt.axes(xlim=(xmin, xmax), ylim=(float(ymin), float(ymax)))
ax.set_title(graph_name)
ax.set_xlabel("Input")
ax.set_ylabel("Output")
```

Pro vytvoření grafu v reálném čase je nutné použít balíček "animation" z "matplotlib":

```
Class matplotlib.animation.FuncAnimation(fig, func, frames=None,
init_func=None, fargs=None, save_count=None, **kwargs)
```

6.4.4 Výsledný blokový diagram programu pro předzpracování dat.

Výsledný blokový diagram programu pro předzpracování dat je zobrazen na obrázku 6.4. Kód programu je vložen do přílohy 1. Program je schopen přijmout data v různých formátech: float, int, short, double. Použitá verze Pythonu je 3.5.

Chování programu je znázorněno na obrázku 6.7. Funkčnost programu byla úspěšně otestována na vstupních datech různých typů: double, float, int, short. Program je schopen přijmout data ve formě pole bajtů, což znamená, že zápis dat na sériový port musí probíhat takto:

```
Serial.write(buf, 20);
```

kde „buf“ je bajtové pole a 20 je jeho velikost. Pokud data budou odeslána jinak, například:

```
Serial.write(A1);
```

```
Serial.write(A2);
```

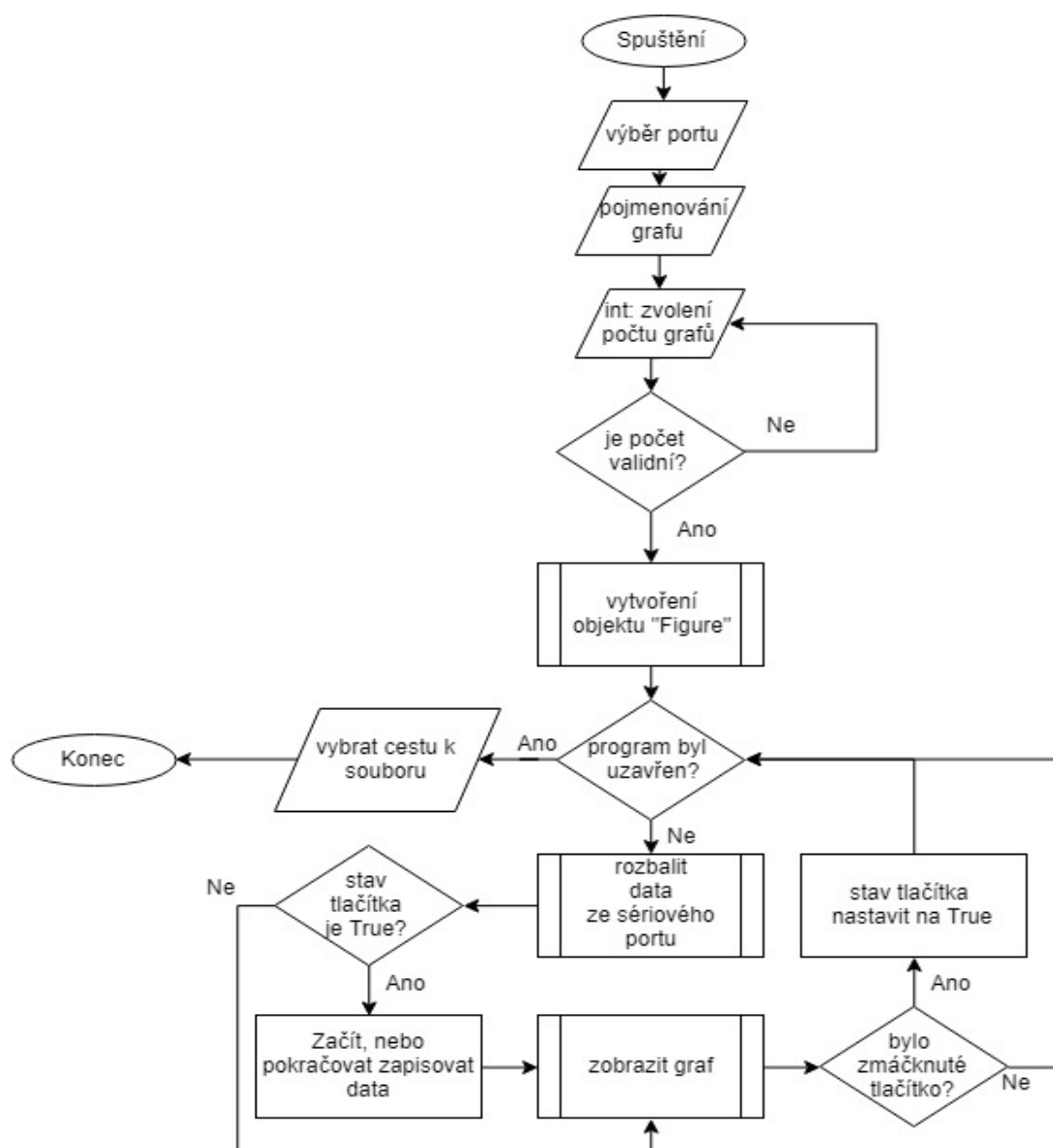
```
Serial.write(A3);
```

dojde k nesprávnému chování programu.

Momentálně je program schopen současně zobrazit maximálně 15 průběhů, avšak omezení spočívá jen v počtu stylů. Díky bohaté funkcionalitě knihovny matplotlib je ovšem snadné přidat velké množství stylů. V programu je to řešeno takto:

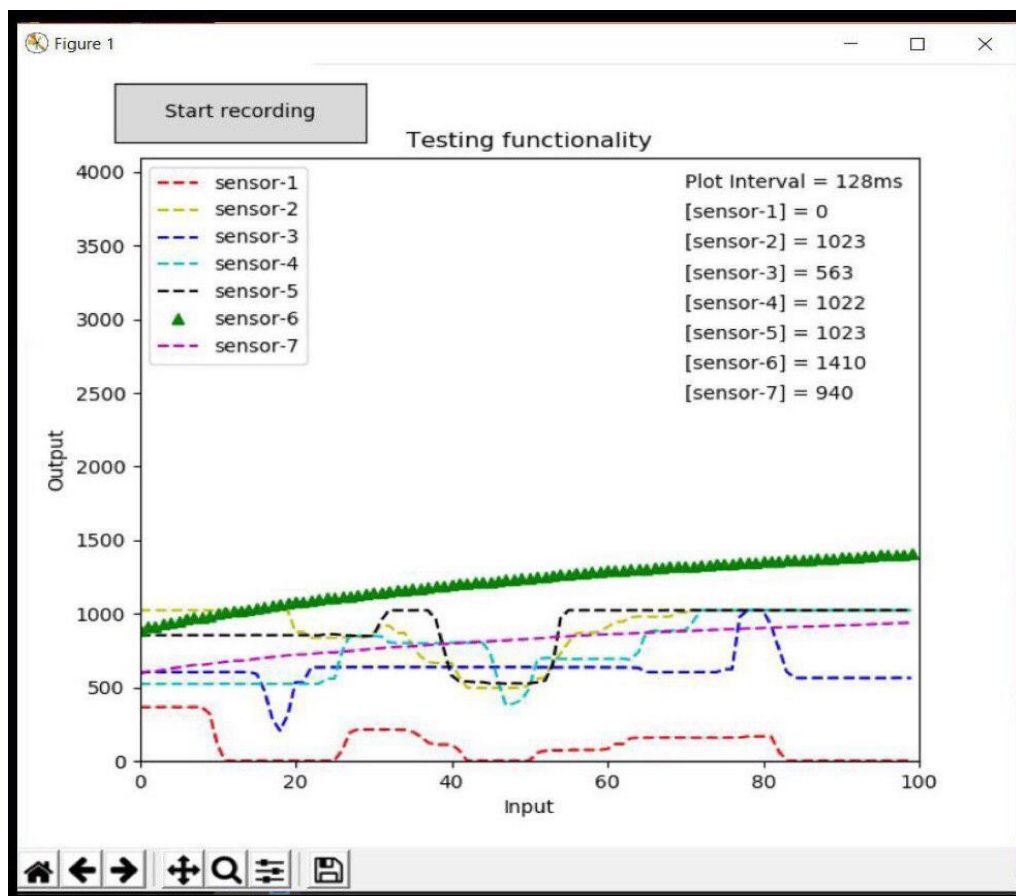
```
style = ['r--', 'y--', 'b--', 'c--', 'k--', 'g^', 'm--', 'r^', 'y^',
'b^', 'c^', 'k^', 'g^', 'm^', 'r:'];
```

kde proměnná „style“ je pole stylů. Každý styl je definován dvěma proměnnými. První proměnná je barva datové značky, například r – červený (red), y – žlutý (yellow) atd. Kromě použití těchto zkratk lze barvu nastavit použitím celého názvu nebo kódu barvy. Druhá proměnná odpovídá za styl datových značek, například „- -“ znamená, že značky budou ve formě čárkovaných čar, „^“ – ve formě trojúhelníků, „:“ – ve formě tečkovaných čar. Seznam všech stylů je k dispozici v dokumentaci matplotlib (dostupné z <https://matplotlib.org>).



Obrázek 6.6: Zjednodušený blokový diagram programu pro předzpracování dat

Po uzavření programu dojde k otevření dialogových oken pro výběr cesty, kam uložit soubory. Uložená data jsou oddělena mezi sebou znakem středník (;). Příklady uložených souborů se nacházejí v příloze na CD.



Obrázek 6.7: Testování funkčnosti programů pro předzpracování dat

6.5 Realizace a testování programu pro zpracování dat pomocí PCA

6.5.1 Knihovny Pandas, Scikit-learn, matplotlib jako základ programu

Knihovna Pandas umožňuje jednoduše manipulovat s velkým množstvím dat a tato data analyzovat. Základním prvkem Pandas je objekt s názvem DataFrame. Jedná se o dvoudimenzionální tabulkovou strukturu, do které lze zapisovat textové a numerické hodnoty. DataFrame obsahuje funkce pro snadnou manipulaci s daty včetně zápisu a čtení dat ve formě tabulek ze souboru, vymazání, přejmenování, nahrazení sloupců nebo řádků v tabulce, nahrazení prázdných hodnot nebo jejich odstranění, třídění dat. V tabulce 6.1 jsou popsány statistické funkce knihovny Pandas.

Tabulka 6.1: Statistické funkce knihovny Pandas

Funkce	Popis
count()	Počet nenulových pozorování

Funkce	Popis
sum()	Součet hodnot
mean()	Průměr hodnot
median()	Aritmetický medián hodnot
min()	Minimum
max()	Maximum
std()	Směrodatná odchylka
var()	Rozptyl
skew()	Koeficient šikmosti
kurt()	Koeficient špičatosti
quantile()	Vzorový kvantil (hodnota v %)
cov()	Kovariance
corr()	Korelace

Pro čtení dat z .csv souboru a na jejich základě vytvoření objektu DataFrame byla použita funkce `read_csv`:

```
df = pd.read_csv(path, sep=';', encoding='utf8');
```

kde `path` je cesta k souboru, `sep` je oddělovač, pomocí kterého jsou oddělená data v souboru.

Pro standardizaci dat byla použita funkce `StandardScaler()`. Každý datový bod je vycentrován pomocí odečítání střední hodnoty a následně je rozdělen směrodatnou odchylkou, jak bylo popsáno v rovnici 11. V tabulce 6.2 jsou zobrazena data před a po standardizaci, kompletní tabulka je dostupná v příloze na CD.

Tabulka 6.2: Prvních pět dat ze senzorů před standardizací (vlevo) a po standardizaci (vpravo)

index	Sensor1	Sensor2	Sensor3	➔	Sensor1	Sensor2	Sensor3
1	306	591	96		-1,1792	-2,2113	-0,9894
2	344	588	101		-0,8877	-2,2351	-0,8669
3	414	918	127		-0,3508	0,3811	-0,2303
4	379	888	119		-0,6193	-0,1433	-0,4262
5	370	625	108		-0,6883	-1,9417	-0,6556

Dalším krokem je dekompozice dat pomocí PCA. V programu to bylo provedeno pomocí funkce `PCA`, která je součástí knihovny Scikit-learn. Při použití této funkce je nutné zadat počet komponent, v tomto případě se jedná o dvě komponenty, protože za pomoci komponent se data zobrazí na dvourozměrném prostoru. Příklad použití:


```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
```

Následně jsou hlavní komponenty uloženy do DataFrame takto:

```
principalDf = pd.DataFrame(data=principalComponents,
columns=['principal component1 ', 'principal component 2'])
finalDf = pd.concat([principalDf, df[['target']], axis=1)
```

V tabulce 6.3 je možno vidět, jak vypadají část hlavních komponent. Kompletní tabulka je dostupná v příloze na CD.

Tabulka 6.3: Příklad dat po převedení do hlavních komponent

index	Hlavní komponenta 1	Hlavní komponenta 2	Cíl
1	-2,4187	1,1839	Vzduch
2	-2,1744	1,13435	Vzduch
3	-0,1727	-0,5302	Rumová
4	-0,5756	-0,4814	Rumová
5	-1,8023	1,2152	Vzduch

Každý datový bod na grafu musí mít svou barvu, pro získání všech barev definovaných v knihovně matplotlib je nutné použít tento řádek kódu:

```
colors = dict(mcolors.BASE_COLORS, **mcolors.CSS4_COLORS)
```

Protože některé barvy nelze rozlišit na běžných obrazovkách, byly vybrány jen některé přísně definované barvy:

```
colors = ['black', 'gray', 'lightgrey', 'indianred', 'red', 'coral',
'chocolate', 'orange', 'olive', 'yellow', 'lawngreen', 'green', 'cyan',
'blue', 'purple']
```

Posledním krokem je vykreslení bodů na grafu pomocí funkce knihovny *matplotlib* `Axes.scatter()`:

```
for target in target_color.keys():
    indicesToKeep = finalDf['target'] == target
    ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1'],
               finalDf.loc[indicesToKeep, 'principal component 2'],
               c=target_color.get(target),
               s=50)
```

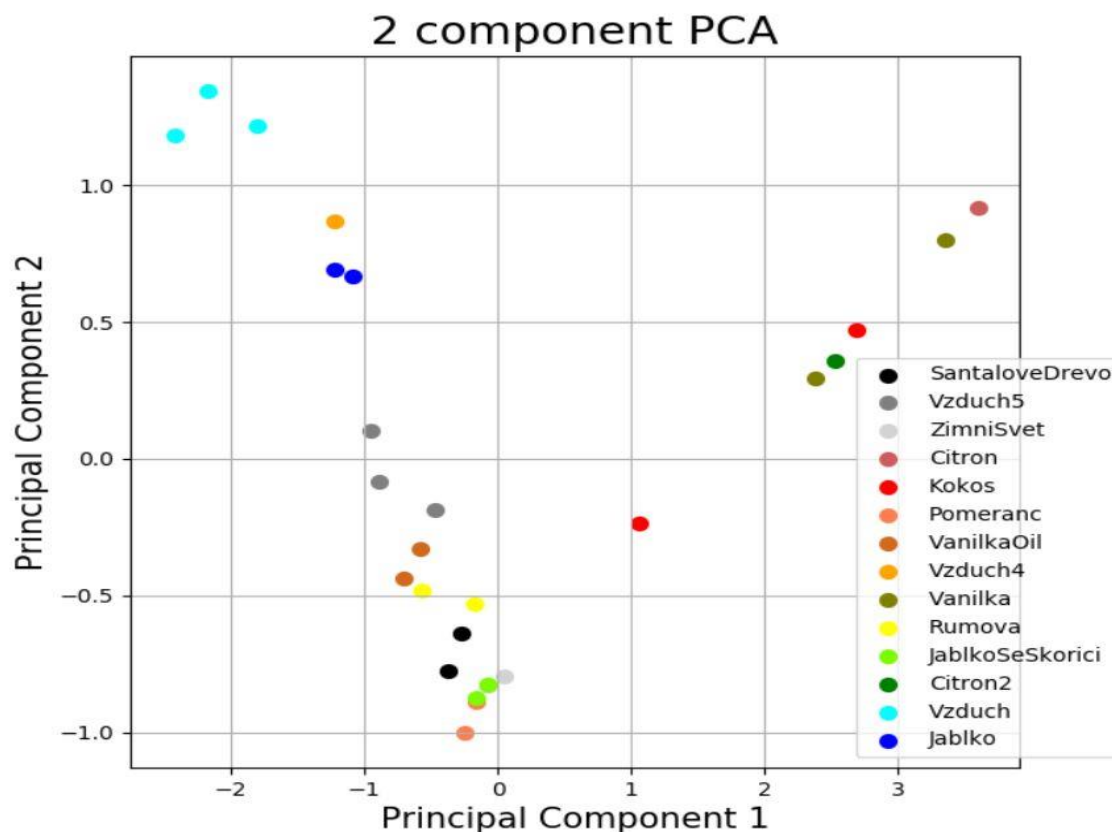
6.6 Testování programu zpracování dat na reálných datech

Pro testování programu byla použita data pana Bc. Beneše, která byla získána pomocí jeho vlastního E-nosu. Pro své zařízení používal tři sensory: MQ-6, který je citlivý na propan nebo isobutan (300 – 10000 ppm), MQ-3 je citlivý na alkohol (25 – 500 ppm) a MQ-8 pro detekci vodíku (100 – 1000 ppm) [33]. Ve své práci pan Bc. Beneš testoval zařízení na potravinářských aromatech nebo na vonných olejích. Seznam použitých vzorků je v tabulce 6.4.

Tabulka 6.4: Seznam použitých vzorků

Název	Výrobce / distributor
Kokosové	Aroco s.r.o.
Vanilkové	Aroco s.r.o.
Citronové	Aroco s.r.o.
Rumové	Dr.Oetker s.r.o.
Pomeranč	Lidl ČR v.o.s
Jablko	Kozák Svitavy s.r.o.
Vanilka	Lidl ČR v.o.s.
Santalové dřevo	Kozák Svitavy s.r.o.
Zimní svět	Lidl ČR v.o.s.
Jablko se skořicí	Lidl ČR v.o.s.

Na obrázku 6.8 je zobrazeno okno aplikace s výsledným bodovým grafem. V grafu je vidět, že vůně, jako kokos, citron, vanilka, pomeranč, nelze klasifikovat pomocí pouhých tří senzorů, protože se mezi sebou protínají a vykazují v grafu velký rozptyl. Jiné vzorky, jako jsou vzduch, santalové dřevo, jablko, jablko se skořicí, vanilkový olej, rumová, jsou potenciálně identifikovatelné, ale abychom jejich identifikovatelnost s určitostí potvrdili, bylo by třeba provést podstatně vyšší počet měření.



Obrázek 6.8: Výsledný bodový graf po zpracování dat

6.7 Výhody a nevýhody použitého řešení

Hlavní výhody jsou:

- jednoduché zapojení, ovládání a programování,
- jazyk Python je zdarma a obsahuje užitečné nástroje pro zpracování dat,
- schopnost modifikovat a zlepšovat navržený elektronický nos.

Hlavní nevýhoda spočívá v tom, že program pro zpracování dat není schopen automatické identifikace látek. Výsledky vizualizace dat musí být analyzovány člověkem.

7. ZÁVĚR

Nejčastěji používanou metodou pro zpracování vícerozměrných dat ze senzorů v potravinářském průmyslu je PCA. PCA slouží k redukci vícerozměrného prostoru, proto se používá spolu s jinými metodami pro zpracování dat. Méněrozměrný prostor zvýší rychlost metody, ale úspěch algoritmu v takovém případě závisí na rekonstrukční chybě. Malý rozptyl hodnot způsobí větší rekonstrukční chybu, proto v případě malého rozptylu hodnot nelze data úspěšně klasifikovat. Standardní PCA vždy nalezne lineární základní komponenty, které reprezentují data v prostoru s nižším rozměrem. Někdy potřebujeme nelineární hlavní komponenty, proto PCA není univerzální metodou. V potravinářství však pomocí PCA lze klasifikovat velké množství různých potravin, což bylo popsáno v teoretické části bakalářské práce.

V praktické části byl navržen systémový blokový diagram pro zpracování dat z elektronického nosu. V popsaném blokovém diagramu mikrokontrolér sbírá data ze sensorového pole a posílá je do počítače přes USB port, kde jsou data zapisována do .csv souboru pro další zpracování a vizualizaci.

Program pro předzpracování umožní čtení dat ze sériového portu a jejich správné uložení do .csv souborů. Pro uložení ustálených dat do souboru bylo přidáno tlačítko, pomocí kterého lze spustit zápis hodnot. Po uzavření programu jsou data ve správném formátu zapsána do dvou souborů, první soubor obsahuje pouze střední hodnoty pro každý senzor, druhý soubor obsahuje všechna data pro případ, kdy střední hodnota nedostačuje k popisu naměřených dat.

Program pro zpracování dat vyžaduje, aby data v souboru byla rozdělena středníkem, nebo aby hodnoty z každého senzoru byly zapisovány zvlášť do různých sloupců v tabulce. Hlavní nevýhodou programu je však to, že program není schopen automatické identifikace látek. Výsledky vizualizace dat musejí být analyzovány člověkem.

Program pro zpracování dat byl napsán v jazyce Python s použitím knihoven Pandas, Scikit-learn, matplotlib, easygui. Vysoká úroveň abstrakce, jednoduchá syntaxe a bohatost na open source knihovny jazyka Python umožňují jednoduché rozšíření funkčnosti aplikací, například přidání dalších metod zpracování dat a automatizaci identifikace vzorků vůní.

Program pro zpracování dat byl otestován na datech získaných panem Bc. Benešem na svém elektronickém nosu, který obsahuje tři senzory. Malý počet senzorů omezuje detekční rozsah látek, což je vidět na obrázku 6.8 v místech, kde se mezi sebou protínají body různých vzorků. Pan Bc. Beneš ve své práci uvedl, že jeho E-nos má měřicí komoru z tenkého plastu, která snadno absorbuje měřené vůně, což mohlo způsobit to, že některé vzorky se objevily v různých místech. Tento problém je dobře patrný například u vzorků pojmenovaných jako vzduch a vzduch5. Potenciálně by šlo identifikovat všechny vzorky, které se neprotínají, ale pro důkladné ověření této domněnky je nutné otestovat program pro větší počet měření.

Dalším pokračováním práce by mohlo spočívat v otestování programu na zařízení s vyšším počtem senzorů. V případě nedostatečné identifikace vzorků pomocí zařízení s vyšším počtem senzorů je nutné zjistit, zda jsou použité senzory vhodné a dostatečně citlivé pro využití v potravinářství. Následné kroky by mohly vést k přidání a následné ověření dalších metod pro testování dat a automatizaci zařízení.

Literatura

- [1] ELUC. *Elektronická učebnice* [online]. Dostupné z:
<https://eluc.kr-olomoucky.cz/verejne/lekce/249>
- [2] GUTIERREZ-OSUNA, R., Jacek GĘBICKI, Waldemar WARDENCKI a Jacek NAMIEŚNIK, 2002. Pattern analysis for machine olfaction: a review. *IEEE Sensors Journal*. **2**(3), 189-202. DOI: 10.1109/ISEN.2002.800688. ISSN 1530-437X. Dostupné z: <http://ieeexplore.ieee.org/document/1021060/>
- [3] KIANI, Sajad, Saeid MINAEI, Mahdi GHASEMI-VARNAMKHAISTI a James A. COVINGTON, 2016. A portable electronic nose as an expert system for aroma-based classification of saffron. *Chemometrics and Intelligent Laboratory Systems*. **156**(3), 148-156. DOI: 10.1016/j.chemolab.2016.05.013. ISSN 01697439. Dostupné z:
<https://linkinghub.elsevier.com/retrieve/pii/S0169743916301228>
- [4] DYMERSKI, Tomasz, Jacek GĘBICKI, Waldemar WARDENCKI a Jacek NAMIEŚNIK, 2014. Application of an Electronic Nose Instrument to Fast Classification of Polish Honey Types. *Sensors*. **14**(6), 10709-10724. DOI: 10.3390/s140610709. ISSN 1424-8220. Dostupné z: <http://www.mdpi.com/1424-8220/14/6/10709>
- [5] KALAW, Justine M. a Fortunato B. SEVILLA III, 2018. Discrimination of wood species based on a carbon nanotube/polymer composite chemiresistor array. *Holzforschung*. **72**(3), 215-223. DOI: 10.1515/hf-2017-0097. ISSN 1437-434X. Dostupné z: <http://www.degruyter.com/view/j/hfsg.2018.72.issue-3/hf-2017-0097/hf-2017-0097.xml>
- [6] RUTOLO, Massimo F., Daciana ILIESCU, John P. CLARKSON a James A. COVINGTON, 2016. Early identification of potato storage disease using an array of metal-oxide based gas sensors. *Postharvest Biology and Technology*. **116**(3), 50-58. DOI: 10.1016/j.postharvbio.2015.12.028. ISSN 09255214. Dostupné z:
<https://linkinghub.elsevier.com/retrieve/pii/S092552141530209X>
- [7] ARGYRI, Anthoula, Efstathios PANAGOU, George-John NYCHAS, E.Z. PANAGOU a G.-J.E. NYCHAS, 2014. Multispectral image analysis approach to detect adulteration of beef and pork in raw meats. *Novel Food Preservation and Microbial Assessment Techniques*. CRC Press, 2014-04-15, **67**, 386-434. DOI: 10.1201/b16758-17. ISBN 978-1-4665-8075-6. ISSN 09639969. Dostupné z:
<http://www.crcnetbase.com/doi/abs/10.1201/b16758-17>
- [8] FU, Xiaping, Moon S. KIM, Kuanglin CHAO, et al., 2014. Detection of melamine in milk powders based on NIR hyperspectral imaging and spectral similarity analyses: a review. *Journal of Food Engineering*. **124**(3), 97-104. DOI: 10.1016/j.jfoodeng.2013.09.023. ISSN 02608774. Dostupné z:
<https://linkinghub.elsevier.com/retrieve/pii/S0260877413004937>
- [9] ROPODI, A.I., E.Z. PANAGOU a G.-J.E. NYCHAS. Data mining derived from food analyses using non-invasive/non-destructive analytical techniques; determination of food authenticity, quality & safety in tandem with computer science disciplines. *Trends in Food Science & Technology*. 2016, **50**, 11-25 DOI:

- 10.1016/j.tifs.2016.01.011. ISSN 09242244. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0924224415301424>
- [10] ROPODI, A.I., D.E. PAVLIDIS, F. MOHAREB, E.Z. PANAGOU a G.-J.E. NYCHAS, 2015. Multispectral image analysis approach to detect adulteration of beef and pork in raw meats. *Food Research International*. CRC Press, 2014-04-15, **67**, 12-18. DOI: 10.1016/j.foodres.2014.10.032. ISBN 978-1-4665-8075-6. ISSN 09639969. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0963996914006887>
- [11] QIU, Shanshan, Jun WANG, Liping GAO, E.Z. PANAGOU a G.-J.E. NYCHAS, 2014. Discrimination and Characterization of Strawberry Juice Based on Electronic Nose and Tongue: Comparison of Different Juice Processing Approaches by LDA, PLSR, RF, and SVM. *Journal of Agricultural and Food Chemistry*. CRC Press, 2014-04-15, **62**(27), 6426-6434. DOI: 10.1021/jf501468b. ISBN 978-1-4665-8075-6. ISSN 0021-8561. Dostupné z: <http://pubs.acs.org/doi/10.1021/jf501468b>
- [12] ZAKARIA, Ammar, Ali Yeon Md SHAKAFF, Maz Jamilah MASNAN, et al., 2012. Improved Maturity and Ripeness Classifications of Magnifera Indica cv. Harumanis Mangoes through Sensor Fusion of an Electronic Nose and Acoustic Sensor: Comparison of Different Juice Processing Approaches by LDA, PLSR, RF, and SVM. *Sensors*. CRC Press, 2014-04-15, **12**(5), 6023-6048. DOI: 10.3390/s120506023. ISBN 978-1-4665-8075-6. ISSN 1424-8220. Dostupné z: <http://www.mdpi.com/1424-8220/12/5/6023>
- [13] WILSON, Alphas, Charisse OBERLE, Daniel OBERLE, et al., 2013. Detection of Off-Flavor in Catfish Using a Conducting Polymer Electronic-Nose Technology: Comparison of Different Juice Processing Approaches by LDA, PLSR, RF, and SVM. *Sensors*. CRC Press, 2014-04-15, **13**(12), 15968-15984. DOI: 10.3390/s131215968. ISBN 978-1-4665-8075-6. ISSN 1424-8220. Dostupné z: <http://www.mdpi.com/1424-8220/13/12/15968>
- [14] PAPADOPOULOU, Olga S., Efstathios Z. PANAGOU, Fady R. MOHAREB, et al., 2013. Sensory and microbiological quality assessment of beef fillets using a portable electronic nose in tandem with support vector machine analysis: Comparison of Different Juice Processing Approaches by LDA, PLSR, RF, and SVM. *Food Research International*. CRC Press, 2014-04-15, **50**(1), 241-249. DOI: 10.1016/j.foodres.2012.10.020. ISBN 978-1-4665-8075-6. ISSN 09639969. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0963996912004310>
- [15] PANAGOU, E.Z., N. SAHGAL, N. MAGAN, et al., 2008. Table olives volatile fingerprints: Potential of an electronic nose for quality discrimination. *Sensors and Actuators B: Chemical*. CRC Press, 2014-04-15, **134**(2), 902-907. DOI: 10.1016/j.snb.2008.06.038. ISBN 978-1-4665-8075-6. ISSN 09254005. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0925400508004516>
- [16] CONCINA, I., M. FALASCONI, E. GOBBI, et al., 2009. Early detection of microbial contamination in processed tomatoes by electronic nose: Potential of an electronic nose for quality discrimination. *Food Control*. CRC Press, 2014-04-15, **20**(10), 873-880. DOI: 10.1016/j.foodcont.2008.11.006. ISBN 978-1-4665-8075-6. ISSN 09567135. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0956713508003174>

- [17] PAN, Leiqing, Wei ZHANG, Na ZHU, et al., 2014. Early detection and classification of pathogenic fungal disease in post-harvest strawberry fruit by electronic nose and gas chromatography–mass spectrometry: Potential of an electronic nose for quality discrimination. *Food Research International*. CRC Press, 2014-04-15, **62**(10), 162-168. DOI: 10.1016/j.foodres.2014.02.020. ISBN 978-1-4665-8075-6. ISSN 09639969. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0963996914001306>
- [18] ZHANG, Hongmei, Jun WANG, Sheng YE, et al., 2012. Application of Electronic Nose and Statistical Analysis to Predict Quality Indices of Peach: Comparison of Different Juice Processing Approaches by LDA, PLSR, RF, and SVM. *Food and Bioprocess Technology*. CRC Press, 2014-04-15, **5**(1), 65-72. DOI: 10.1007/s11947-009-0295-7. ISBN 978-1-4665-8075-6. ISSN 1935-5130. Dostupné z: <http://link.springer.com/10.1007/s11947-009-0295-7>
- [19] DEISEMRPTH, Marc Peter, A Aldo FAISAL a Cheng Soon ONG. Mathematics for Machine Learning [online]. Cambridge University Press, 2019 [cit. 2019-05-25]. Dostupné z: <https://mml-book.github.io/>
- [20] CASALINUOVO, Ida, Donato DI PIERRO, Massimiliano COLETTA a Paolo DI FRANCESCO, 2006. Application of Electronic Noses for Disease Diagnosis and Food Spoilage Detection. *Sensors*. **6**(11), 1428-1439. DOI: 10.3390/s6111428. ISSN 1424-8220. Dostupné z: <http://www.mdpi.com/1424-8220/6/11/1428>
- [21] KONVALINA, Gady, Hossam HAICK, Orna BARASH, Hossam HAICK a Marcis LEJA, 2013. Sensors for Breath Testing: From Nanomaterials to Comprehensive Disease Detection. *Accounts of Chemical Research*. **47**(1), 66-76. DOI: 10.1021/ar400070m. ISSN 0001-4842. Dostupné z: <http://pubs.acs.org/doi/10.1021/ar400070m>
- [22] TISCH, Ulrike, Hossam HAICK, Orna BARASH, Hossam HAICK a Marcis LEJA, 2010. Arrays of chemisensitive monolayer-capped metallic nanoparticles for diagnostic breath testing: From Nanomaterials to Comprehensive Disease Detection. *Reviews in Chemical Engineering*. **26**(5-6), 66-76. DOI: 10.1515/revce.2010.009. ISSN 2191-0235. Dostupné z: <https://www.degruyter.com/view/j/revce.2010.26.issue-5-6/revce.2010.009/revce.2010.009.xml>
- [23] ESTAKHROYEH, Hossein Rezaei, Esmat RASHEDI, Mahdijeh MEHRAN, Hossam HAICK a Marcis LEJA, 2018. Design and Construction of Electronic Nose for Multi-purpose Applications by Sensor Array Arrangement Using IBGSA: From Nanomaterials to Comprehensive Disease Detection. *Reviews in Chemical Engineering*. **92**(2), 205-221. DOI: 10.1007/s10846-017-0759-3. ISSN 0921-0296. Dostupné z: <http://link.springer.com/10.1007/s10846-017-0759-3>
- [24] Wiring [online]. [cit. 2019-05-21]. Dostupné z: <http://wiring.org.co>
- [25] Wiring Framework. Wiring [online]. [cit. 2019-05-25]. Dostupné z: <http://wiring.org.co/reference/>
- [26] PRECHELT, Lutz, 2003. *Are Scripting Languages Any Good? A Validation of Perl, Python, REXX, and Tcl against C, C++, and Java*. Elsevier, 2003, 205-270. Advances in Computers. DOI: 10.1016/S0065-2458(03)57005-X. ISBN 9780120121571. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S006524580357005X>

- [27] OLIPHANT, Travis E., 2007. *Python for Scientific Computing*. **9**(3), 10-20. DOI: 10.1109/MCSE.2007.58. ISSN 1521-9615. Dostupné z: <http://ieeexplore.ieee.org/document/4160250/>
- [28] SHUKLA, Xitij U. a Dinesh J. PARMAR, 2016. Python – A comprehensive yet free programming language for statisticians. *Journal of Statistics and Management Systems*. **19**(2), 277-284. DOI: 10.1080/09720510.2015.1103446. ISSN 0972-0510. Dostupné z: <http://www.tandfonline.com/doi/full/10.1080/09720510.2015.1103446>
- [29] PySerial. *PySerial* [online]. [cit. 2019-05-25]. Dostupné z: <https://pythonhosted.org/pyserial/>
- [30] Struct – Interpret bytes as packed binary data. *Python* [online]. 2019 [cit. 2019-05-25]. Dostupné z: <https://docs.python.org/3/library/struct.html>
- [31] Easygui 0.98.1. *PyPI* [online]. [cit. 2019-05-25]. Dostupné z: <https://pypi.org/project/easygui/>
- [32] *Построение графиков в Python при помощи Matplotlib* [online]. [cit. 2019-05-25]. Dostupné z: <https://python-scripts.com/matplotlib>
- [33] BENEŠ, D. Přenosné zařízení pro detekci vůně v potravinářství. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky, 2016. 63 s. Bakalářská práce. Vedoucí práce: Ing. Martin Adámek, Ph.D.

Seznam příloh

Příloha 1 - Obsah přiloženého CD.	43
--	----

Příloha 1 - Obsah přiloženého CD.

Složka *předzpracování* obsahuje spustitelný program pro předzpracování dat, pro ho testování je nutné připojit testovací zařízení do sériového portu. Složka *zdrojové kódy* obsahuje všechny zdrojové kódy včetně *simulation.ino*, který byl použit pro programování mikrokontroléru a pro testování programu předzpracování dat. Ve složce *zpracování pomocí PCA* se nachází spustitelný program pro zpracování dat, program byl otestován na datech ze souboru *pca2.csv*. Ve hlavní složce se nachází plné soubory dat, které jsou částečně ukázané v bakalářské práci.